```
RRRRRRRRRRR      MMM          MMM     SSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSSSS
RRR         RRR  MMMMMM    MMMMMM  SSS
RRR         RRR  MMMMMM    MMMMMM  SSS
RRR         RRR  MMMMMM    MMMMMM  SSS
RRR         RRR  MMM   MMM   MMM   SSS
RRR         RRR  MMM   MMM   MMM   SSS
RRR         RRR  MMM   MMM   MMM   SSS
RRRRRRRRRRR      MMM          MMM     SSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSS
RRR   RRR        MMM          MMM             SSS
RRR   RRR        MMM          MMM             SSS
RRR   RRR        MMM          MMM             SSS
RRR      RRR     MMM          MMM             SSS
RRR      RRR     MMM          MMM             SSS
RRR      RRR     MMM          MMM             SSS
RRR         RRR  MMM          MMM  SSSSSSSSSSS
RRR         RRR  MMM          MMM  SSSSSSSSSSS
RRR         RRR  MMM          MMM  SSSSSSSSSSS
```

```
NN      NN  TTTTTTTTTT    000000    DDDDDDDD     IIIIII     SSSSSSSS  PPPPPPPP    LL          YY      YY
NN      NN  TTTTTTTTTT    000000    DDDDDDDD     IIIIII     SSSSSSSS  PPPPPPPP    LL          YY      YY
NN      NN      TT      00      00  DD      DD     II     SS          PP      PP  LL          YY      YY
NN      NN      TT      00      00  DD      DD     II     SS          PP      PP  LL          YY      YY
NNNN    NN      TT      00    0000  DD      DD     II     SS          PP      PP  LL            YY  YY
NNNN    NN      TT      00    0000  DD      DD     II     SS          PP      PP  LL            YY  YY
NN  NN  NN      TT      00  00  00  DD      DD     II       SSSSSS    PPPPPPPP    LL              YY
NN  NN  NN      TT      00  00  00  DD      DD     II       SSSSSS    PPPPPPPP    LL              YY
NN    NNNN      TT      0000    00  DD      DD     II             SS  PP          LL              YY
NN    NNNN      TT      0000    00  DD      DD     II             SS  PP          LL              YY
NN      NN      TT      00      00  DD      DD     II             SS  PP          LL              YY          ....
NN      NN      TT      00      00  DD      DD     II             SS  PP          LL              YY          ....
NN      NN      TT        000000    DDDDDDDD     IIIIII     SSSSSSSS  PP          LLLLLLLLLL      YY          ....
NN      NN      TT        000000    DDDDDDDD     IIIIII     SSSSSSSS  PP          LLLLLLLLLL      YY          ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

(2) 64 DECLARATIONS
(3) 94 NT$DISPLAY - PERFORM NETWORK DISPLAY FUNCTION

```
0000     1          $BEGIN  NTODISPLY,000,NF$NETWORK,<NETWORK DISPLAY>
0000     2
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27  ;
0000    28
0000    29  ;++
0000    30  ; Facility: RMS
0000    31  ;
0000    32  ; Abstract:
0000    33  ;
0000    34  ;       This module communicates with the file access listener at the remote
0000    35  ;       node to display attributes of the specified file.
0000    36  ;
0000    37  ; Environment: VAX/VMS, executive mode
0000    38  ;
0000    39  ; Author: James A. Krycka,      Creation Date:  01-JUN-1981
0000    40  ;
0000    41  ; Modified By:
0000    42  ;
0000    43  ;       V03-003 JAK0134         J A Krycka      04-MAR-1984
0000    44  ;               Correction to V03-001 to properly handle the receipt of a DAP
0000    45  ;               Attributes message without a Name message in the sequence.
0000    46  ;
0000    47  ;       V03-002 JAK0130         J A Krycka      06-JAN-1984
0000    48  ;               Return alternate success of RMS$_OK_NOP if remote system does
0000    49  ;               not support the DAP DISPLAY function instead of an RMS$_SUPPORT
0000    50  ;               error.
0000    51  ;
0000    52  ;       V03-001 KRM0094         K Malik         29-MAR-1983
0000    53  ;               Request Resultant Name message and fill in user Name block
0000    54  ;               (if appropriate).
0000    55  ;
0000    56  ;       V02-003 JAK0066         J A Krycka      07-OCT-1981
0000    57  ;               Reject $DISPLAY request while in file transfer mode.
```

```
0000     58 ;
0000     59 ;       V02-002 JAK0063         J A Krycka      17-AUG-1981
0000     60 ;               Add support for the Revision Date and Time XAB.
0000     61 ;
0000     62 ;--
```

```
0000     64          .SBTTL  DECLARATIONS
0000     65
0000     66 ;
0000     67 ; Include Files:
0000     68 ;
0000     69
0000     70          $DAPPLGDEF                      ; Define DAP prologue symbols
0000     71          $DAPHDRDEF                      ; Define DAP message header
0000     72          $DAPCNFDEF                      ; Define DAP Configuration message
0000     73          $DAPACCDEF                      ; Define DAP Access message
0000     74          $DAPCTLDEF                      ; Define DAP Control message
0000     75          $IFBDEF                         ; Define IFAB symbols
0000     76          $NWADEF                         ; Define Network Work Area symbols
0000     77
0000     78 ;
0000     79 ; Macros:
0000     80 ;
0000     81 ;        None
0000     82 ;
0000     83 ; Equated Symbols:
0000     84 ;
0000     85
0000     86          ASSUME  DAP$Q_DCODE_FLG EQ 0
0000     87
0000     88 ;
0000     89 ; Own Storage:
0000     90 ;
0000     91 ;        None
0000     92 ;
```

```
                     0000     94          .SBTTL  NT$DISPLAY - PERFORM NETWORK DISPLAY FUNCTION
                     0000     95
                     0000     96 ;++
                     0000     97 ; NT$DISPLAY - engages in a DAP dialogue with the remote FAL to display
                     0000     98 ;       attributes of the specified file according to the XABs chained to the
                     0000     99 ;       FAB.
                     0000    100 ;
                     0000    101 ; Calling Sequence:
                     0000    102 ;
                     0000    103 ;       BSBW    NT$DISPLAY
                     0000    104 ;
                     0000    105 ; Input Parameters:
                     0000    106 ;
                     0000    107 ;       R8      FAB address
                     0000    108 ;       R9      IFAB address
                     0000    109 ;       R10     IFAB address
                     0000    110 ;       R11     Impure Area address
                     0000    111 ;
                     0000    112 ; Implicit Inputs:
                     0000    113 ;
                     0000    114 ;       User FAB
                     0000    115 ;       DAP$Q_SYSCAP
                     0000    116 ;       NWA$V_FTM_RETRV
                     0000    117 ;       NWA$V_FTM_STORE
                     0000    118 ;
                     0000    119 ; Output Parameters:
                     0000    120 ;
                     0000    121 ;       R0      Status code (RMS)
                     0000    122 ;       R1-R7   Destroyed
                     0000    123 ;       AP      Destroyed
                     0000    124 ;
                     0000    125 ; Implicit Outputs:
                     0000    126 ;
                     0000    127 ;       User ALL, DAT, FHC, KEY, PRO, RDT, SUM XABs
                     0000    128 ;       User NAM (resultant name string)
                     0000    129 ;
                     0000    130 ; Completion Codes:
                     0000    131 ;
                     0000    132 ;       Standard RMS completion codes
                     0000    133 ;
                     0000    134 ; Side Effects:
                     0000    135 ;
                     0000    136 ;       None
                     0000    137 ;
                     0000    138 ;--
                     0000    139
                     0000    140 NT$DISPLAY::                            ; Entry point
                     0000    141         $TSTPT  NTDISPLAY               ;
57   3C A9   D0     0006    142         MOVL    IFB$L_NWA_PTR(R9),R7    ; Get address of NWA (and DAP)
        11   E1     000A    143         BBC     #DAP$V_DISPLAY,-        ; Branch if remote FAL does not support
  10 28 A7                 000C    144                 DAP$Q_SYSCAP(R7),ALTSUC ;  display attributes function
   67   1A   E0     000F    145         BBS     #NWA$V_FTM_RETRV,(R7),- ; Branch if file transfer mode retrieval
        06           0012    146                 ERRFTM                  ;  in progress
   67   1B   E0     0013    147         BBS     #NWA$V_FTM_STORE,(R7),- ; Branch if file transfer mode storage
        02           0016    148                 ERRFTM                  ;  in progress
        0C   11     0017    149         BRB     BUILD_MASK              ; Proceed
             0019    150 ERRFTM: RMSERR  FTM                             ; Declare file transfer mode error
```

```
                 05  001E   151                 RSB                             ; Exit with RMS code in R0
                     001F   152 ALTSUC:  RMSERR  OK_NOP                          ; Return alternate success code
                 05  0024   153                 RSB                             ; Exit with RMS code in R0
                     0025   154
                     0025   155 ;+
                     0025   156 ; Build display field request mask which will be used in the Control message
                     0025   157 ; to request that optional DAP messages be returned by FAL. For $DISPLAY these
                     0025   158 ; are the ATT, ALL, KEY, PRO, SUM, and TIM messages. (Note that the Attributes
                     0025   159 ; message will supply information to update the FHCXAB).
                     0025   160 ;
                     0025   161 ;-
                     0025   162
                     0025   163 BUILD_MASK:                                      ; Build NWA$W_DISPLAY
                   56  D4   0025   164                 CLRL    R6                  ; Indicate this is not a close operation
                 FFD6'  30   0027   165                 BSBW    NT$SCAN_XABCHN      ; Scan user XAB chain and check FAL's
                     002A   166                                                     ;  capabilities; request mask put in R2
                03 50  E8   002A   167                 BLBS    R0,5$               ; Branch on success
                  0085  31   002D   168                 BRW     FAIL                ; Branch (failed to complete scan)
             20  00C6 C7  91   0030   169 5$:          CMPB    NWA$B_ORG(R7),#NWA$K_IDX; Branch if IDX organization
                    03  13   0035   170                 BEQL    10$                 ;
                 52  0A  AA   0037   171                 BICW2   #<<DAP$M_DSP_SUM>!- ; Ignore these XABs in chain
                     003A   172                           <DAP$M_DSP_KEY>!-         ;
                     003A   173                           0>,R2                     ;
                   FFC3'  30   003A   174 10$:          BSBW    NT$SCAN_NAMBLK      ; Scan user NAM block and check
                75 50  E9   003D   175                 BLBC    R0,FAIL             ;  FAL's capabilities; update
                     0040   176                                                     ;  request mask
             00D0 C7  52  B0   0040   177                 MOVW    R2,NWA$W_DISPLAY(R7); Save request mask
                    6B  13   0045   178                 BEQL    SUC                 ; Exit if there are no XABs to fill
                     0047   179
                     0047   180 ;+
                     0047   181 ; Build and send DAP Control message to partner.
                     0047   182 ;-
                     0047   183
                     0047   184 SEND_CTL:                                        ; (required message sequence)
                     0047   185                 $SETBIT #NWA$V_LAST_MSG,(R7)    ; Declare this last message to block
                 50  04  D0   004B   186                 MOVL    #DAP$K_CTL_MSG,R0   ; Get message type value
                   FFAF'  30   004E   187                 BSBW    NT$BUILD_HEAD       ; Construct message header
                 85  10  90   0051   188                 MOVB    #DAP$K_DISPLAY,(R5)+; Store CTLFUNC field
                 85  20  90   0054   189                 MOVB    #DAP$M_DISPLAY2,(R5)+; Store CTLMENU field
             51  00D0 C7  3C   0057   190                 MOVZWL  NWA$W_DISPLAY(R7),R1; Get request mask
                   FFA1'  30   005C   191                 BSBW    NT$CVT_BN4_EXT      ; Store DISPLAY as an extensible field
                   FF9E'  30   005F   192                 BSBW    NT$BUILD_TAIL       ; Finish building message
                   FF9B'  30   0062   193                 BSBW    NT$TRANSMIT         ; Send Control message to FAL
                  4D 50   E9   0065   194                 BLBC    R0,FAIL             ; Branch on failure
                     0068   195
                     0068   196 ;+
                     0068   197 ; Receive DAP Attributes message from partner and update the user FHCXAB.
                     0068   198 ;-
                     0068   199
                     0068   200 RECV_ATT:                                        ; (optional--must be requested)
                     0068   201                 BBC     #DAP$V_DSP_ATT,-        ; Branch if Attributes message was not
                 00D0 C7      006A   202                 NWA$W_DISPLAY(R7),-     ;  requested
                     0E       006D   203                 RECV_EXT_ATT            ;
                     006E   204                 $SETBIT #DAP$K_ATT_MSG,DAP$L_MSG_MASK(R7)
                     0073   205                                                     ; Expect response of Attributes message
                   FF8A'  30   0073   206                 BSBW    NT$RECEIVE          ; Get reply from FAL
                  3C 50   E9   0076   207                 BLBC    R0,FAIL             ; Branch on failure
```

```
                FF84'  30   0079   208          BSBW    NT$UPDATE_FHC           ; Update user FHCXAB
                       0079   209
                       0079   210 ;+
                       0079   211 ; Receive DAP Extended Attributes messages from partner and update the user
                       0079   212 ; ALL, DAT, KEY, PRO, RDT, and SUM XABs.
                       0079   213 ;-
                       0079   214
                       0079   215 RECV_EXT_ATT:                               ; (optional--must be requested)
                FF81'  30   0079   216          BSBW    NT$RECV_EXT_ATT         ; Process Extended Attributes messages
              33 50    E9   007C   217          BLBC    R0,FAIL                 ; Branch on failure
                       007F   218
                       007F   219 ;+
                       007F   220 ; Receive DAP Resultant Name message from partner.
                       007F   221 ;-
                       007F   222
                       007F   223 RECV_NAME:                                   ; (optional--must be requested)
                   08  E1   007F   224          BBC     #DAP$V_DSP_NAM,-        ; Branch if Name message was not
                00D0 C7      0081   225                  NWA$W_DISPLAY(R7),-     ;  requested
                     1F      0084   226                  RECV_ACK                ;
                             0085   227          $SETBIT #DAP$K_NAM_MSG,DAP$L_MSG_MASK(R7)
                             008A   228                                          ; Expect response of Name message
                FF70'  30   008A   229          BSBW    NT$RECEIVE              ; Get reply from FAL
              22 50    E9   008D   230          BLBC    R0,FAIL                 ; Branch on failure
           5A 38 A9    D0   0090   231          MOVL    IFB$L_FWA_PTR(R9),R10   ; Get FWA address (NT$DECODE_NAM needs
                FF66'  30   0094   232          BSBW    NT$DECODE_NAM           ;  it) & process resultant name string
                FF63'  30   0097   233          BSBW    RM$FILLNAM              ; Fill in user NAM block (destroys R7)
              15 50    E9   009A   234          BLBC    R0,FAIL                 ; Branch on failure
           57 3C A9    D0   009D   235          MOVL    IFB$L_NWA_PTR(R9),R7    ; Restore NWA address
              5A 59    D0   00A1   236          MOVL    R9,R10                  ; Restore IFAB address
                       00A4   237
                       00A4   238 ;+
                       00A4   239 ; Receive DAP Acknowledge message from partner.
                       00A4   240 ;-
                       00A4   241
                       00A4   242 RECV_ACK:                                    ; (required message)
                              00A4   243          $SETBIT #DAP$K_ACK_MSG,DAP$L_MSG_MASK(R7)
                              00A9   244                                       ; Expect response of Acknowledge message
                FF51'  30   00A9   245          BSBW    NT$RECEIVE              ; Get reply from FAL
              03 50    E9   00AC   246          BLBC    R0,FAIL                 ; Branch on failure
                       00AF   247 SUC:    RMSSUC                              ; Return success
                    05 00B2   248 FAIL:   RSB                                 ; Exit with RMS code in R0
                       00B3   249
                       00B3   250          .END                                ; End of module
```

```
$$.PSECT_EP                     = 00000000
$$RMSTEST                       = 0000001A
$$RMS_PBUGCHK                   = 00000010
$$RMS_TBUGCHK                   = 00000008
$$RMS_UMODE                     = 00000004
ALTSUC                            0000001F  R      01
BUILD_MASK                        00000025  R      01
DAP$B_ACCFUNC                     00000040
DAP$B_ACCOPT                      00000041
DAP$B_BITCNT                      00000035
DAP$B_BLKCNT                      00000056
DAP$B_CTLFUNC                     00000040
DAP$B_DCODE_FID                   00000019
DAP$B_DCODE_MAC                   0000001B
DAP$B_DCODE_MSG                   0000001A
DAP$B_DECVER                      00000047
DAP$B_ECONUM                      00000045
DAP$B_FAC                         00000042
DAP$B_FILESYS                     00000043
DAP$B_FLAGS                       00000031
DAP$B_KRF                         00000047
DAP$B_LEN256                      00000034
DAP$B_LENGTH                      00000033
DAP$B_OSTYPE                      00000042
DAP$B_RAC                         00000046
DAP$B_SHR                         00000043
DAP$B_STREAMID                    00000032
DAP$B_TYPE                        00000030
DAP$B_USRNUM                      00000046
DAP$B_USRVER                      00000048
DAP$B_VERNUM                      00000044
DAP$B_X_FIELD                     00000024
DAP$C_BLN                         000000C0
DAP$K_ACK_MSG                   = 00000006
DAP$K_ATT_MSG                   = 00000002
DAP$K_BLN                         000000C0
DAP$K_CTL_MSG                   = 00000004
DAP$K_DISPLAY                   = 00000010
DAP$K_NAM_MSG                   = 0000000F
DAP$K_SEQ_ACC                   = 00000000
DAP$L_CMWA                        00000030
DAP$L_CRC_RSLT                    00000020
DAP$L_DCODE_STS                   00000018
DAP$L_MSG_MASK                    0000001C
DAP$L_ROP                         00000050
DAP$L_SSPWA                       00000080
DAP$L_TEMP                        00000090
DAP$M_BITCNT                    = 00000008
DAP$M_BLKCNT                    = 00000040
DAP$M_DISPLAY2                  = 00000020
DAP$M_DSP_3NAM                  = 00000200
DAP$M_DSP_KEY                   = 00000002
DAP$M_DSP_SUM                   = 00000008
DAP$M_GET                       = 00000002
DAP$M_GO_NOGO                   = 00000010
DAP$M_MSE                       = 00000010
DAP$M_SEGMENT                   = 00000040
DAP$M_TMP1$                     = 00000008
DAP$M_TMP2$                     = FFF80000
DAP$Q_DCODE_FLG                   00000000
DAP$Q_FILESPEC                    00000044
DAP$Q_KEY                         00000048
DAP$Q_MSG_BUF1                    00000008
DAP$Q_MSG_BUF2                    00000010
DAP$Q_PASSWORD                    00000050
DAP$Q_SYSCAP                      00000028
DAP$Q_SYSPEC                      00000038
DAP$V_DISPLAY                   = 00000011
DAP$V_DSP_ATT                   = 00000000
DAP$V_DSP_NAM                   = 00000008
DAP$W_BUFSIZ                      00000040
DAP$W_CTLMENU                     00000044
DAP$W_DISPLAY1                    0000004C
DAP$W_DISPLAY2                    00000054
DAP$W_PARTNER                     00000006
DAP$W_VERSION                     00000004
ERRFTM                            00000019  R      01
FAIL                              000000B5  R      01
IFB$L_FWA_PTR                   = 00000038
IFB$L_NWA_PTR                   = 0000003C
NT$BUILD_HEAD                     ********    X    01
NT$BUILD_TAIL                     ********    X    01
NT$CVT_BN4_EXT                    ********    X    01
NT$DECODE_NAM                     ********    X    01
NT$DISPLAY                        00000000  RG     01
NT$RECEIVE                        ********    X    01
NT$RECV_EXT_ATT                   ********    X    01
NT$SCAN_NAMBLK                    ********    X    01
NT$SCAN_XABCHN                    ********    X    01
NT$TRANSMIT                       ********    X    01
NT$UPDATE_FHC                     ********    X    01
NWA$B_ALLXABCNT                   0000011C
NWA$B_DAP_RAC                     000000C9
NWA$B_FILESYS                     000000C5
NWA$B_KEYXABCNT                   0000011D
NWA$B_NETSTRSIZ                   0000016F
NWA$B_NODBUFSIZ                   00000168
NWA$B_ORG                         000000C6
NWA$B_OSTYPE                      000000C4
NWA$B_RFM                         000000C7
NWA$B_RMS_RAC                     000000C8
NWA$C_BLN                         00000800
NWA$K_BLN                         00000800
NWA$K_IDX                       = 00000020
NWA$L_ALLXABADR                   00000100
NWA$L_DATXABADR                   00000104
NWA$L_DEV                         000000C0
NWA$L_FHCXABADR                   00000108
NWA$L_KEYXABADR                   0000010C
NWA$L_MSG_MASK                    000000D4
NWA$L_PROXABADR                   00000110
NWA$L_RDTXABADR                   00000114
NWA$L_SAVE_FLGS                   00000128
NWA$L_SUMXABADR                   00000118
```

```
NWA$L_THREAD                    000000FC
NWA$L_XLTATTR                   00000238
NWA$L_XLTBUFFLG                 0000022C
NWA$L_XLTCNT                    00000228
NWA$L_XLTMAXINDX                00000234
NWA$L_XLTSIZ                    00000230
NWA$Q_ACS                       00000244
NWA$Q_BIGBUF                    00000170
NWA$Q_BLD                       000000F0
NWA$Q_FLG                       00000000
NWA$Q_INODE                     0000025C
NWA$Q_IOSB                      000000D8
NWA$Q_LNODE                     00000160
NWA$Q_LOGNAME                   0000023C
NWA$Q_NCB                       00000264
NWA$Q_RCV                       000000E0
NWA$Q_SAVE_DESC                 00000120
NWA$Q_XLTBUF1                   0000024C
NWA$Q_XLTBUF2                   00000254
NWA$Q_XMT                       000000E8
NWA$T_ACSBUF                    0000026C
NWA$T_AUXBUF                    000005E0
NWA$T_DAP                       00000000
NWA$T_INODEBUF                  000004AC
NWA$T_ITM_ATTR                  00000200
NWA$T_ITM_END                   00000224
NWA$T_ITM_LST                   00000200
NWA$T_ITM_MAXINDX               00000218
NWA$T_ITM_STRING                0000020C
NWA$T_NCBBUF                    0000052C
NWA$T_NODEBUF                   00000169
NWA$T_RCVBUF                    000001A0
NWA$T_SCAN                      00000100
NWA$T_TEMP                      00000120
NWA$T_XLTBUF1                   000002AC
NWA$T_XLTBUF2                   000003AC
NWA$T_XMTBUF                    000003C0
NWA$V_FTM_RETRV               = 0000001A
NWA$V_FTM_STORE               = 0000001B
NWA$V_LAST_MSG                = 00000000
NWA$W_BUILD                     000000D2
NWA$W_DAPBUFSIZ                 000000CA
NWA$W_DIR_OFF                   000000CC
NWA$W_DISPLAY                   000000D0
NWA$W_FIL_OFF                   000000CE
NWA$W_JNLABJOP                  0000011E
PIO$A_TRACE                     ********    X   01
RECV_ACK                        000000A7  R     01
RECV_ATT                        00000068  R     01
RECV_EXT_ATT                    0000007C  R     01
RECV_NAME                       00000082  R     01
RMS$FILLNAM                     ********    X   01
RMS$_FTM                        ********    X   01
RMS$_OK_NOP                     ********    X   01
SEND_CTL                        00000047  R     01
SUC                             000000B2  R     01
TPT$L_NTDISPLAY                 ********    X   01
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
NF$NETWORK	000000B6 (182.)	01 (1.)	PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC BYTE
ABS	00000800 (2048.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.09	00:00:00.96
Command processing	125	00:00:00.60	00:00:03.08
Pass 1	237	00:00:07.93	00:00:21.49
Symbol table sort	0	00:00:00.95	00:00:01.29
Pass 2	59	00:00:01.37	00:00:03.93
Symbol table output	21	00:00:00.16	00:00:00.27
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	478	00:00:11.12	00:00:31.08

The working set limit was 1350 pages.
37804 bytes (74 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 665 non-local and 6 local symbols.
250 source lines were read in Pass 1, producing 13 object records in Pass 2.
22 pages of virtual memory were used to define 21 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[RMS.OBJ]RMS.MLB;1	13
_255DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	17

895 GETS were required to define 17 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:NTODISPLY/OBJ=OBJ$:NTODISPLY MSRC$:NTODISPLY/UPDATE=(ENH$:NTODISPLY)+LIB$:RMS/LIB

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

NT0DAPRMS
LIS

NT0GET
LIS

NT0NWASET
LIS

NT0EXTEND
LIS

NT0DECODE
LIS

NT0ENCODE
LIS

NT0ERASE
LIS

NT0DISCON
LIS

NT0DISPLY
LIS

NT0MISC
LIS